Faster-RCNNを用いた名刺項目の位置推定と識別

Faster-RCNN を用いた名刺項目の位置推定と識別

POSITION ESTIMATION AND RECOGNITION OF BUSINESS CARD ITEMS USING FASTER-RCNN

西尾　一磨
Kazuma NISHIO
指導教員 木村　光宏

法政大学大学院理工学研究科システム理工学専攻修士課程

In this study, we try to construct a deep-learning-based discriminator that can recognize the information on a business card. In particular, the discriminator can locate where the card holder's data (it may comprize his/her name, address, phone number, email, etc.) are on the card, and recognize what each information is. Our research consists of two approaches and models. The first model extracts HOG (Histogram of Oriented Gradients) feature quantities that are already located on the card, and classifies each HOG feature into several prespecified labels (name, address, and so on). In the second model, we modify our first model to recognize the location where each label is on the card by utilizing the methodology of the Faster RCNN (region convolutional neural network). In this thesis, we develop the above two models and investigate the discrimination performance for the sample data sets.
***Key Words** : Faster RCNN, HOG, image recognition, deep learning*

1.　はじめに

従来，国内でやり取りされる名刺データの管理は OCR(Optical Character Recognition) 技術を使用し名刺内の情報を読み取っているが, その正確性は 8 割程度である 1). 名刺のデータ情報は 1 文字の読み取りができないだけでも大きな問題になる (例えば電話がつながらないなど). そのため名刺管理のほとんどは手作業で行われることが多いのが現状である．本研究では，深層学習を用いて名刺画像のピクセル値データから，名刺のどこにどんな情報が記載されているかについて，位置推定と予測を高精度に行うマルチタスクな学習モデルの生成を目指す.

2.　データセット

本研究ではデータセットとして Sansan 株式会社が提供している実在しないサンプル名刺の画像データを使用する**??**. この画像データは実際に名刺をスキャナやカメラで取り込んだものである． 学習用に 983 枚, バリデーション用データとして 120 枚，テスト用に 100 枚を PNG 形式で使用した. 名刺内の項目とその領域の位置座標の情報を取得するため「labelImg」という Github に公開されているアノテーションツールを用いた．名刺項目は「会社名」「氏名」「役職」「住所 (郵便番号を含む)」「電話番号」「FAX 番号」「E-mail アドレス」「HP の URL」のどれか一つの項目をラベル付けし，項目が表示されている領域を四角形の形で区切りその位置座標を取得した．合計領域数は学習用およびバリデーション用 9826 領域，テスト用に 865 領域となった．名刺によっては上記の項目のうちで記載されていないものも存在する．また光の反射などで名刺内の文字が認識しづらいものは削除した.

3.　分析手法

(1)　二つの分析手法

本研究では二つの手法を用いて実験を行う．一つ目は画像データから HOG(Histogram of Oriented Gradients, 以下 HOG と称す) 特徴量を抽出し，抽出した HOG 特徴量を入力データとして畳み込みニューラルネットワーク (Convolutional Neural Network，以下 CNN と称す) を用いて名刺項目の識別を行う．この実験では名刺項目の識別のみを行うため，名刺画像そのものではなく名刺項目の領域を入力データとして使用する．CNN では畳み込み層で特徴量を自動で生成するため，入力画像データをそのまま用いることが多いが，今回は HOG 特徴量変換した後に CNN で学習させた際にどうなるかという実験する．二つ目は Faster Region Convolutional Neural Network（以下 Faster RCNN と称す）という深層学習モデルを用いて名刺画像データから，どこの領域に何の項目が表示されているかを位置推定，識別する．二つ目の手法が本研究の目的を果たすものである．Faster RCNN は 2015 年に提案された物体検出の手法**??**であり，この手法を用いて名刺項目の識別と位置推定を高精度にすることを目指す.

4.　HOG と CNN を用いた名刺項目の識別

以下は一つ目の実験である，HOG と CNN を用いた名刺項目の識別の実験方法について述べていく.

(1)　HOG 特徴量

HOG とは局所領域 (セル) の画素値の勾配方向をヒストグラム化したものである**??**. 画像のデータセットを学習させる際，各ピクセルごとの画素値を情報として学習を行う．本研究ではデータセットの情報を要約して次元削減を試みるため HOG 特徴量変換を行う．これは局所領域 (セル) の画素値の勾配方向をヒストグラム化しこれを特徴量としたものである．HOG の計算アルゴリズムは (a) 輝度勾配の算出，(b) それらの勾配方向を一定の方向ごとにヒストグラム化，(c) 局所ブロックによる正規化，の 3 つの処理から構成される．本稿ではセルのサイズを 12 × 12 ピクセル，ブロックのサイズも 12 × 12 ピクセル，画像サイズは 168 × 84 ピクセルとし

た．輝度勾配は, 式 (1)〜(3) のように算出される．

$$\begin{cases} f_x(x,y) = L(x+1,y) - L(x-1,y) \\ f_y(x,y) = L(x,y-1) - L(x,y+1) \end{cases}, \tag{1}$$

$$m = \sqrt{f_x(x,y)^2 + f_y(x,y)^2}, \tag{2}$$

$$\theta(x,y) = \tan^{-1}\frac{f_y(x,y)}{f_x(x,y)}, \tag{3}$$

$$h'(n) = \frac{h(n)}{H}, \tag{4}$$

$$H = \sqrt{(\sum_{k=1}^{m\times m\times N} h(k)^2) + \epsilon^2}. \tag{5}$$

式 (1) の $L(x,y)$ は画像の座標 (x,y) の輝度値である. 勾配強度 m と勾配方向 θ を用いて 12 × 12 ピクセルをセルとした領域において輝度の勾配のヒストグラムを作成する. 勾配方向ヒストグラムの 0° から 180° を 8 分割し 8 方向のヒストグラムを作成する. 正規化処理は, 各セルで作成した勾配ヒストグラムをセルのサイズに応じて式 (4) により正規化する. $h(n)$ は n 番目の勾配方向ヒストグラムの数値であり, H は 1 ブロックの HOG 特徴量の総和で式 (5) により算出する. m はセルサイズ, N は勾配方向数, また ϵ=1 とした. 特徴量の次元の算出は以下のようになる. Dim は画像全体の HOG 特徴量の次元数, Image width, Image height は入力画像の幅と高さ, Cell size はセルサイズ, Block size はブロックサイズ, Orientation は勾配方向のヒストグラムの方向数である.

$$\begin{aligned} \mathrm{Dim} = &(\frac{\text{Image width}}{\text{Cell size}} - \text{Block size} + 1) \\ &\times (\frac{\text{Image height}}{\text{Cell size}} - \text{Block size} + 1) \\ &\times \text{Block size}^2 \times \text{Orientation}. \end{aligned} \tag{6}$$

(2) 畳み込みニューラルネットワーク (CNN)

畳み込みニューラルネットワークは深層学習の一つであり画像認識分野において利用される手法である [5]. 本研究での CNN は, 入力層 (input layer), 2 つの畳み込み層 (convolution layer), 2 つのプーリング層 (pooling layer), 2 つの全結合層 (fully connected layer), 出力層 (output layer) から構成される (図 1). 入力画像の次元は 784 次元であり, これを 28×28 ピクセルの正方形行列と考え, CNN の入力層のデータとする. 今回は 5×5 ピクセルの特徴マップを 32 個用意した. 特徴マップの初期値は一様分布からサンプリングした. その次に畳み込み層 (1), プーリング層 (1), 畳み込み層 (2), プーリング層 (2), と計算処理をし, 1 つ目の全結合層に続く.1 つ目では, プーリング層 (2) から出力された 4×4×64 の値を 1×1024 の数値に平坦化する. そして, 2 つ目の全結合層では, 9 つの出力となるようなソフトマックス出力層として機能する.

(3) 畳み込み演算

畳み込み演算を $Y = X * W$ で表す. 出力行列を Y, 入力行列を $X_{n1\times n2}$, フィルター行列を $W_{m1\times m2}(m1 \le n1, m2 \le n2)$ とする. これを数学的に定義すると, 式 (7) のようになる.

$$Y[i,j] = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} X[i-k_1, j-k_2]W[k_1,k_2]. \tag{7}$$

この総和では, X と W のインデックス参照の方向が異なるため, フィルタ W を回転させ, 回転したフィルタ W^r として演算を行う.

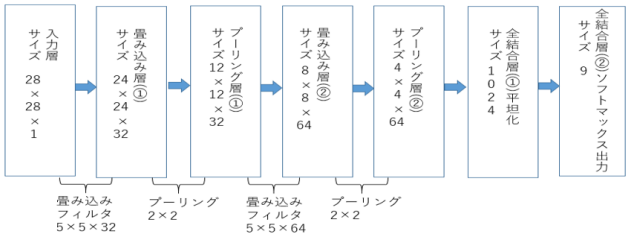

図 1 CNN アーキテクチャ

(4) プーリング

本稿のプーリングは 2 層とも, 最大値プーリング (max-pooling) を用いた. プーリング層を $P_{n1\times n2}$ とする. $n1 \times n2$ は最大値プーリングが実行される近傍のサイズであり, 2×2 とした. 最大値プーリングは, 局所不変性をもたらすため, 入力データのノイズに対し, 頑健な特徴量を生成するのが目的である.

(5) ドロップアウト

ドロップアウトとは, 出力側の隠れ層のユニットで適応され, ニューラルネットワークのトレーニング過程で, ドロップアウト確率 P_{drop} で隠れ層のユニットの一部をランダムに取り除く. 本稿では, ドロップアウト確率を 0.5 とした. これにより, 大規模なニューラルネットワークが過学習に陥るのを防ぐのが目的である.

(6) ソフトマックス出力層

ソフトマックス出力層では, ソフトマックス関数を使用し, 9 つのクラスのインデックスを確率的に指定する. ソフトマックス関数の数式は以下のようになる.

$$P(y=i|z) = \phi(z) = \frac{\exp(z_i)}{\sum_{i=1}^{9}\exp(z_i)}\ (i = 1,2,\ldots,9). \tag{8}$$

z_i は 2 層目全結合層 i 番目の入力ユニットの値である.

(7) 損失関数

損失関数は学習過程で最適化される目的関数であり, この目的関数を最小化することが, 正しく学習できているかの指標となる. 今回は, 交差エントロピー誤差 (cross entropy error) を用いた. 交差エントロピー誤差は以下の式になる.

$$E(w,b) = -\sum_i (t_i \log(y_i) + (1-t_i)\log(1-y_i)). \tag{9}$$

y_i はニューラルネットワークの最終的な出力であり, t_i は正解ラベル, b はバイアスとする. w については次項で述べる.

(8) 勾配降下法

勾配降下法 (gradient descent method) は損失関数を最小化する重みを見つけ出す方法である. 勾配降下法の原理は, 損失関数の数値が極小値または大局的最小値になるまで, 重みを更新する方法である. 勾配降下法では, 次の式を用いて損失関数を最小値へと近づける. なお, η は学習係数である.

$$w \Leftarrow w - \eta\frac{\partial E(w,b)}{\partial w}. \tag{10}$$

5. Faster-RCNN を用いた名刺項目の位置推定と識別

以下は二つ目の実験である，Faster-RCNN を用いた名刺項目の識別と位置推定の実験方法について述べていく．

(1) Faster RCNN の全体構成

入力画像を学習済みの CNN Layer にて演算処理を行い Feature map を抽出する．今回は VGG16 と呼ばれる学習済み CNN モデルを使用し，この CNN モデルから抽出された Feature map の情報を bounding box 提案ネットワーク (Region Proposal Network(RPN と称す)) と呼ばれる物体候補領域 (物体が存在し得る画像領域の候補) を抽出するためのネットワークに入力し，抽出された物体候補領域が物体なのか背景なのか (今回の場合，物体とは名刺画像上にある氏名や住所などが記載されている領域のことを指し，非物体はそれ以外の領域のことである) を学習する．その後物体であると検出された領域に具体的に何が写っているかを学習するという 2 段構造をもたせた．図 2 は Faster RCNN の全体構成の図である．また bounding box とは物体と思われる領域を矩形で囲んだものを指す．

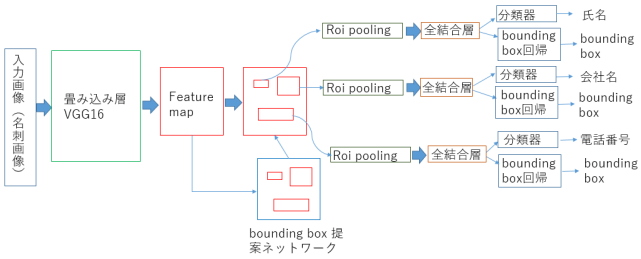

図 2 : Faster RCNN の全体構成例.

(2) Feature map

Feature map とはある入力画像を学習済み CNN モデル (今回は VGG16) を用いて，最後の畳み込み層まで計算した際の出力層のことである．VGG16 とは 13 層からなる畳み込み層と 3 層からなる全結合層のニューラルネットワークのことである．例えば入力層のサイズが 224 × 224 × 3 のとき Feature map のサイズは 7 × 7 × 512 となる (縦横は元画像の 1/32 となる)**??**．なお今回の VGG16 は 1000 枚の名刺画像データから物体となる領域をあらかじめ抽出しクラス識別に，のみによる学習を 100 エポック分行い，学習済み VGG16 モデルを生成した．

(3) RPN の構造

畳み込み層の特徴マップの局所領域ごとに，物体らしさのスコアが付与された複数の bounding box を RPN は提案する**??**．このスコア付き bounding box を提案するために図 3 に示すように bounding box のパラメータを予測する回帰ネットワークと (図 3(b)) 物体の有無を予測する分類ネットワーク (図 3(a)) の 2 つを結合することで，RPN の構造は出来上がる．また，図 3 の Sliding window とはあるきまった大きさの領域を一定のピクセルごとにずらすことで領域候補を提案する際の矩形のことである (本研究では Feature map に対して 3 × 3 の枠を走査した)．

(4) Anchor

Feature map を作成後，次に Anchor の設定を行う．Anchor とは Feature map 上の各点である．例えば下図のように 576 × 800 ピクセルサイズの画像の場合 1 つ当たりの Feature map のサイズは 18 × 25=450 ピクセルとなり，これらの格子状にある点すべてが Anchor となる．

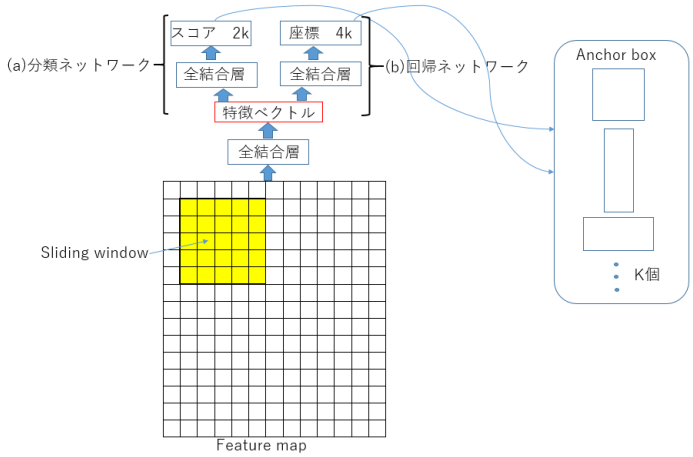

図 3 : RPN の構造.

(5) Anchor box

各 Anchor に対して基準の長さと縦横比をそれぞれ決め複数の Anchor box を作り出す．例えば，基準の長さを 64,128,256 ピクセルとし，縦横比を 1:1，1:2，2:1(正方形，横長矩形，縦長矩形) とした場合，1 つの Anchor に対して，9 個の Anchor box が生成される．また画像からはみ出た Anchor box は無視する．図 4 はある Anchor における Anchor box の作成例である．Anchor boxes を作成する際の注意点として，Anchor boxes の面積は揃えた．例えば，基準の長さを 64 ピクセルにしたとき，Anchor boxes は以下のように整数のサイズで表される．

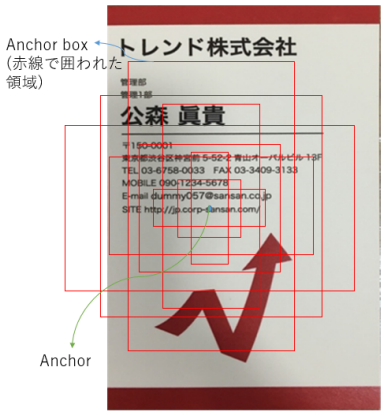

図 4 : Anchor box と Anchor の関係.

1 : 1 → 64 × 64(= 4096),

1 : 2 → 45 × 91(≒ 4096),

2 : 1 → 91 × 45(≒ 4096).

Anchor box において Anchor の間隔や形やサイズを変えることで ground truth(正解となる座標で囲まれた領域) に近い Anchor box を作ることが目的である．

(6) 物体か非物体かの判別

RPN では，Anchor box の領域内の情報が背景か物体かを判別する．これらの情報は図 3 の分類ネットワークにより，1 つの Anchor に対して $2k$ 個 (k は 1 つの Anchor に対する Anchor box の数) のスコアとして出力される．物体か非物体か

の有無は Anchor box と ground truth の IoU(Intersection over Union) という手法を用いて判別する．IoU とは Intersection(領域の共通部分) を Union(領域の和集合) で割った値のこと，すなわち重複する面積の割合である．2 つの矩形が完全に一致していた場合 IoU=1 となり，まったく重なってなかった場合は 0 となる．本研究では，IoU<0.3 の場合は非物体，IoU>0.7 なら IoU は物体，$0.3 \leq \text{IoU} \leq 0.7$ なら学習に使用しないものと定義した．さらに，1 つの ground truth 領域に対して複数の重複する Anchor box がある場合は，Non Maximum Suppression という手法でいづれか 1 つに絞り込む．

(7)　Non maximum Suppression

RPN での検出結果によっては，検出対象の中心として複数の bounding box が検出されてしまうことがある．同一物体に複数の bounding box が検出されないようにするために，bounding box ごとに検出の信頼度を表すスコアを計算し，局所的に最大スコアの bouding box のみを「表示」とし，その他を「非表示」として抑制する．この処理のことを Non Maximum Suppression(NMS と称す) という**??**.

(8)　RoI(Region of Interest) プーリング

RPN により出力された物体候補領域の幅と高さはそれぞれ異なっているため，各領域候補の特徴マップの次元数も異なる．全結合層は，決まった次元数のベクトルしか入力できないため，次元数の異なる Feature map を直接利用できない．そこで，RoI プーリングを使用して次元数の異なる Feature map を一定の大きさのベクトルに変換する**??**．本研究では RoI プーリングのサイズを 7×7 の固定サイズにとした．まず図 5 のように物体候補領域の座標を整数値に変換する．また，図 6 の例は幅 14，高さ 9 の物体候補領域が出力された例を表している．幅を $\frac{14}{7} = 2$，高さを $\frac{9}{7} \fallingdotseq 1.29 = 1$(小数点切り捨て) とし，これらのサイズで区切ったセルの中で MaxPooling を行い 7×7 サイズの RoI Pooling に変換する．RoI プーリング後のベクトルは全結合層を通じて RoI 特徴ベクトルに変換される．RoI 特徴ベクトルは N_c クラスの事後確率を出力する全結合ネットワークと bounding box への回帰ネットワークそれぞれに入力される．

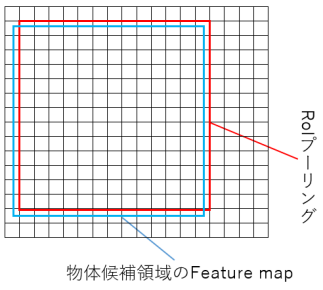

図 5 : 物体候補領域の座標を整数値に変換．

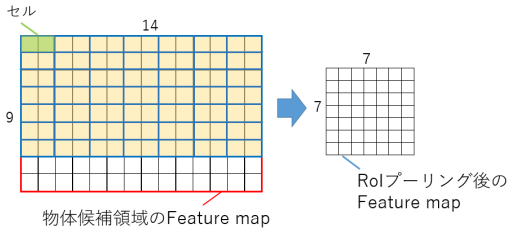

図 6 : RoI プーリング (7×7) の例．

(9)　分類ネットワークと回帰ネットワーク

クラスの事後確率を予測するための分類ネットワークは，物体の有無の 2 クラスを Anchor box ごとに判断するので，$2k$ 次元のベクトル $p^i = (p^0, ..., p^{2k})^{\mathrm{T}}$ を出力する．bounding box への回帰ネットワークは，各 Anchor box からの相対的な位置と相対サイズを含んだ $4k$ 次元のベクトルを出力する．このとき (x, y, w, h) を予測した bounding box(x, y は bounding box の中心座標，w, h は幅と高さ)，(x_a, y_a, w_a, h_a) を Anchor box，(x^*, y^*, w^*, h^*) を ground truth の位置情報として，以下の式の $(t_x^i, t_y^i, t_w^i, t_h^i)$ を Anchor box と予測した bounding box との相対位置および相対サイズ，$t_x^{*i}, t_y^{*i}, t_w^*, t_h^{*i})$ を Anchor box と ground truth の相対位置および相対サイズである．

$$t_x^i = (x - x_a)/w_a, \tag{11}$$

$$t_y^i = (y - y_a)/h_a, \tag{12}$$

$$t_w^i = \log(w/w_a), \tag{13}$$

$$t_h^i = \log(h/h_a), \tag{14}$$

$$t_x^{*i} = (x^* - x_a)/w_a, \tag{15}$$

$$t_y^{*i} = (y^* - y_a)/h_a, \tag{16}$$

$$t_w^{*i} = \log(w^*/w_a), \tag{17}$$

$$t_h^{*i} = \log(h^*/h_a). \tag{18}$$

(10)　学習方法と損失関数

Faster RCNN ではクラス認識と bounding box への回帰を同時に学習するために，ground truth(正解 bounding box) ごとに計算されるマルチタスク損失 (multi-task loss) を最適化することでネットワークのパラメータを求める．学習は以下の損失関数 $L(p^i, p^{*i}, t^i, t^*)$ を最小化するように確率的勾配降下法と誤差逆伝播法を用いて行う．

$$L(p^i, p^{*i}, t^i, t^{*i}) = \alpha \frac{1}{N_{class}} \sum_i L_{class}(p^i, p^{*i}) + \beta \frac{1}{N_{regression}} \sum_i p^{*i} L_{regression}(t^i, t^{*i}) \tag{19}$$

N_{class} は学習におけるミニバッチサイズ，$N_{regression}$ は CNN 特徴マップに対する走査枠の個数である．p^{*i} は検出対象が物体であると判断した場合に $p^{*i} = 1$，非物体であると判断した場合に $p^{*i} = 0$ となる．p^i は i 番目の Anchor box の数である．

$$L_{class}(p^i, c) = -log p_c^i. \tag{20}$$

$L_{class}(p^i)$ は正解ラベル $c \in 0, 1$ のクラス確率 p_c^i に対する対数損失であり，クラス分類に関する損失関数である．

$$L_{regression}(t^i, t^{*i}) = \sum_{j \in x,y,w,h} SmoothL1(t_j^i - t_j^{*i}) \tag{21}$$

$$SmoothL1(x) = \begin{cases} 0.5x^2 & (if |x| < 1) \\ |x| - 0.5 & (otherwise). \end{cases} \tag{22}$$

$L_{regression}(t^i, t^{*i})$ は Anchor box ごとの出力 t^i と t^{*i} の差に対する矩形回帰損失である．α, β は $L_{class}(p^i,)$ と $L_{reg}(t^i, t^{*i})$ の重みを調節するパラメータである．これらの損失関数を最小化することによって，RPN は入力された Feature map 上の Anchor box に対する物体確率と矩形回帰を出力する．

6. 実験結果

本研究では HOG と CNN モデルを用いた名刺項目の識別 (実験 1) と，Faster-RCNN モデルを用いた名刺項目の位置推定とクラス識別を同時に行うマルチタスク学習 (実験 2) の 2 つの実験結果を示す．動作環境は OS:Windows 10 Pro, CPU:Intel(R) Xeon(R) CPU E5-1603v4 @2.80GHz, メモリ:32.0GB である．また，開発環境は Jupyter Notebook にて Python3(version 3.6) とした．

(1) 実験 1 の結果

今回は 50 エポックのトレーニングが完了した後, 学習済みのモデルを保存復元し再び 50 エポックのトレーニングを行った．それにより, モデルをその都度再トレーニングする必要がなくなり, 計算時間が短縮されるのではないかと考えた．しかしながら，初めの 50 エポックの計算時間は 4373.5 秒であるのに対し，次の 50 エポックの計算時間は 4482.2 秒であった．この 2 つの 50 エポック分の学習トレーニングに対する損失関数のグラフを図 7, 8 に示す．どちらのグラフもエポックが増えるごとに減少傾向にあるため，学習トレーニングが上手く行っている可能性が高い．テストデータの名刺項目の正答率は 91.02%と高水準な結果となった．

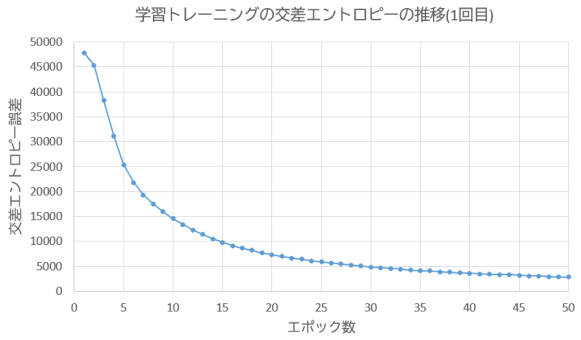

図 7 : 実験 1: 1 回目の損失関数のグラフ.

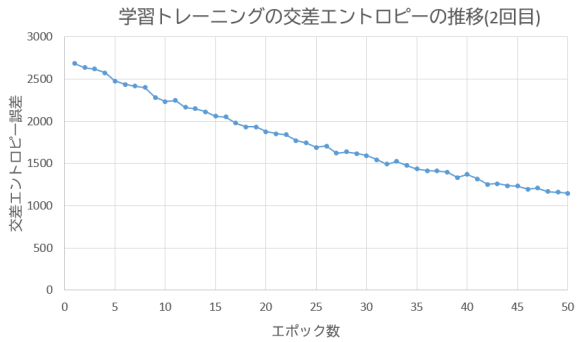

図 8 : 実験 2: 2 回目の損失関数のグラフ.

(2) 実験 2 の評価方法と結果

a) パラメータ

実験 2 では Anchor box のサイズと縦横比 (表 1)，マルチタスク損失関数の α, β の数値表 (2) を変えることにより，名刺項目の位置推定とクラス識別の精度を向上させることを目的とした．本研究では 5 つの実験モデルを作りモデル性能の検証を行う．すべての実験で 5000 エポックの学習を行った．

b) 評価方法

予測した bounding box の領域を R_p とし，正解となる bounding box(ground truth) の領域を R_g とする．それらの一致度合いを定量化する必要がある．よって以下の式を満たすときに物体検出の再現ができたものとする．

$$\frac{R_p \cap R_g \text{の面積}}{R_p \cup R_g \text{の面積}} > 0.5. \tag{23}$$

c) 実験 2 の結果

下の表に 5 つの実験についての Anchor box の設定，損失関数のバランスパラメータ，性能比較の表をまとめる．

表 1 Anchor box の設定.

実験名	サイズ	縦横比 (width:height)
実験 2-1	64^2, 128^2, 256^2	(1:1),(1:3),(3:1)
実験 2-2	64^2, 128^2, 256^2	(1:1)(,1:3),(3:1)
実験 2-3	64^2, 128^2, 256^2, 512^2	(1:1),(1:3),(3:1)
実験 2-4	64^2, 128^2, 256^2, 512^2	(1:1),(1:4),(4:1)
実験 2-5	64^2, 128^2, 256^2, 512^2	(1:1),(1:4),(4:1)

表 2 損失関数の α と β の値

実験名	α	β
実験 2-1	1.0	1.0
実験 2-2	1.0	1.0
実験 2-3	1.0	5.0
実験 2-4	1.0	10.0
実験 2-5	5.0	1.0

表 3 各実験の性能比較 (テストデータより).

実験名	物体再現率	物体クラス認識率
実験 2-1	0.929	0.836
実験 2-2	0.843	0.878
実験 2-3	0.926	0.903
実験 2-4	0.901	0.894
実験 2-5	0.704	0.897

物体再現率は，先で述べた評価方法より物体として検出された bounding box の検出領域が ground truth 検出領域と共通面積領域が 0.5 を超え，かつ bounding box のと ground trurh のクラス認識が正しく一致していた場合の割合を示している．物体クラス認識率とは物体として検出されたすべての bounding box の確信度の値の平均を実験ごとに表した確率である．物体再現率が最も高かったのは実験 2-1 となった．これは NMS の閾値が高く設定したため，予測した bounding box の数が他の実験よりも多くなってしまったのが原因であると考えられる．本来は 1 つの ground truth に対し 1 つの bounding box が出力されるのが望ましいとしているため，実験 1 の結果は不適切であると判断した．その次に物体再現率が高かったのは実験 2-3 であり，物体クラス認識率もすべての実験の中で最も高い．よって本研究での名刺項目の識別と位置推定のモデルに最も適したモデルであると考える．

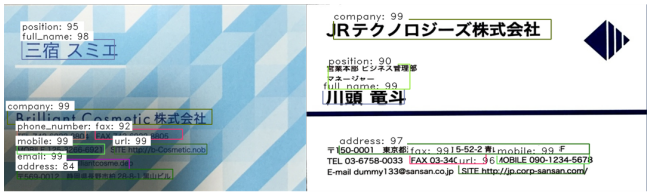

図 9 : 実験 2-3 によるテスト画像の出力例.

図 9 は実験 2-3 のテスト画像の例である．右の画像はすべての項目に 1 つの bounding box が生成されており，物体検出が上手くいった例である．右の画像は TEL から始まる電話番号と E-mail の項目領域が正しく検出されなかった例である．図 10 は実験 2-3 のバリデーションデータによる損

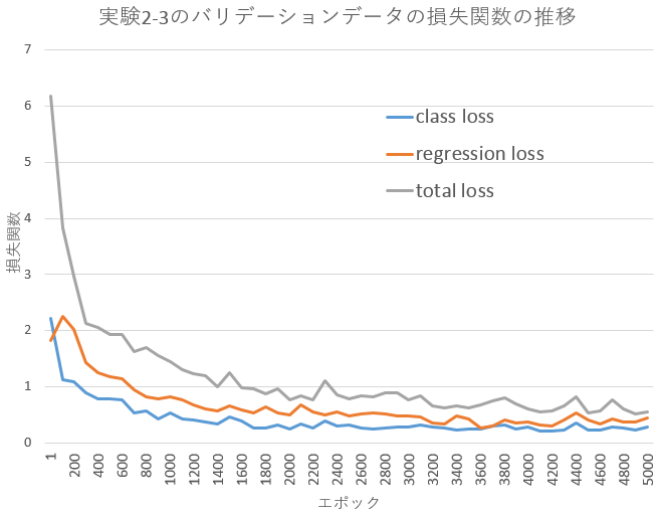

図 10 : 実験 2-3 の損失関数グラフ.

失関数の推移である．class loss は式の (19) 右辺の第 1 項の数値，regression loss は式 (19) 右辺第 2 項の数値，total loss は式 (19) の左辺である．図 10 ではエポックが増えるごとに徐々に減少し 0 の値に近づいていることが読み取れる．

7. 考察

実験 1 の結果より，テスト名刺画像の項目識別の精度が 91.02%となった．この結果から HOG 特徴量を抽出した後 CNN で学習させることは名刺画像の項目の識別において有効な方法ではないかと考える．実験 2 では, 名刺項目の識別と位置推定のモデルを構築し，その中で高精度なモデルを作ることを目標にし実験を行った．どの実験結果にも図 8 の右画像のように識別されない項目領域が存在した．物体再現率を上げるためには，Anchor の数を増やしたり，Anchor box のサイズを変更することが有効ではないかと考える．多様な Anchor box をつくることで，さまざまな物体候補を抽出することができる．また，本研究で使用した回帰ネットワークにおける損失関数は予測した bounding box の中心座標と矩形の幅，高さのズレを学習した．この過程で，IoU などの共通部分の面積の割合などを損失関数の要素として学習してみるとどうなるのかを実験することも高精度な物体検出に有用であると考える. また，実験 2 ではクラス分類損失と矩形回帰損失にバランスパラメータにて調節を行い，マルチタスク損失関数の依存度の変化を研究した．実験 2-3 から矩形回帰損失の比重を上げることで，学習性能が高くなるという結論に至った．だが，実験 2-4 のように比重を上げすぎると逆に学習性能が下がってしまったため，バランスの良い比重を見つけることで学習性能を向上できるのではないかと考える．また，実験 2-3 では，class loss と regression loss の値が徐々に近くなっており，total loss の数値に与える影響が同じくらいである．結果として，この状況のときの物体再現率や物体クラス認識率が他の実験よりも高かったため，class loss と regression loss の値を近づけることは高性能なマルチタスク学習モデルを生成するのに必要な指標ではないかと考える．

参考文献

1) https://enterprisezine.jp/dbonline/detail/8334 (2019 年 10 月閲覧).
2) 国立情報学研究所 (NII) 情報学研究データリポジトリ (Informatics Research Data Repository: IDR) Sansan(https://www.nii.ac.jp/dsc/idr/sansan/) (2019 年 6 月閲覧).
3) S. Ren, K. He, R. Girshick, and J. Sun: “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, arXiv: 1506.01497 (2015).
4) 大西 克則, 滝口 哲也, 有木 康雄: 「HOG 特徴に基づく単眼画像からの人体 3 次元姿勢推定」画像の認識・理解シンポジウム (MIRU2008).
5) 斎藤康毅: ゼロから作る Deep Learning, オライリージャパン，オーム社 (2018).
6) https://www.codetd.com/ja/article/7266381 (2019 年 12 月閲覧).
7) 原田達也:MLP 機械学習プロフェッショナルシリーズ 画像認識 (2017) 講談社サイエンティフィック.
8) K. He, X. Zhang, S. Ren, J. Sun: “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”, arXiv: 1406.4729 (2014).